Hepatoprotective effects of systemic ER activation

ChIPseq/Epigenome genome - Enhancer-gene pair analysis

Christian Sommerauer & Carlos Gallardo

15 September, 2022

```
library(tidyverse)
```

Use BEDOPS suite to determine the closest TSS to the enhancer sites. Run this in terminal, may adjust paths. Requires BEDOPS and bedtools.

```
#Import the closest genes as determined by
closest <- read.delim("results/Epigenome_analysis/origin_closest1_left_closest_1_2_right.bed", header=F
enhancer_ID <- c(1:1816)
# Row 738 has an NA in several columns (including V8), remove this one as it causes issues downstream.
closest2 <- closest %>%
separate(col = V3, into = c("V3_left", "V3_right"), sep = "\\|") %>%
separate(col = V5, into = c("V5_left", "V5_right"), sep = "\\|") %>%
separate(col = V7, into = c("V7_left", "V7_right"), sep = "\\|") %>%
mutate(enhancer_ID=enhancer_ID) %>%
filter(!is.na(V8))

closest3 <- closest2 %>% mutate(enha.ident = paste0(V1, ".",V2,".",V3_left), .before = V1)

colnames(closest3) <- c("enha.ident", "ori_chrom", "ori_start", "ori_end", "left_1_chrom", "left_1_star
                        "right_1_chrom", "right_1_start", "right_1_end",
                        "right_2_chrom", "right_2_start", "right_2_end", "enhancer_ID")


  closest_ori <- closest3 %>% dplyr::select(1, 14, 2:4) %>% mutate(query_ID = "closest_ori")
  colnames(closest_ori) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")
  closest_left1 <- closest3 %>% dplyr::select(1, 14, 5:7) %>% mutate(query_ID = "closest_left1")
  colnames(closest_left1) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")

  closest_right1 <- closest3 %>% dplyr::select(1, 14, 8:10) %>% mutate(query_ID = "closest_right1")
  colnames(closest_right1) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")
  closest_right2 <- closest3 %>% dplyr::select(1, 14, 11:13) %>% mutate(query_ID = "closest_right2")
  colnames(closest_right2) <-  c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")

closest_long <- rbind(closest_ori, closest_left1,
                      closest_right1, closest_right2)
```

```
#subset the enhancer_df to have the exact locations in three columns for later
location <- closest_ori %>% dplyr::select(1,3,4,5)
```

Annotate the closest genes

```
library("ChIPpeakAnno")
library("GenomicRanges")
options(connectionObserver = NULL) #That is a work-around, as the org.Mm. package cannot be loaded
library("org.Mm.eg.db")
library("biomaRt")

gr_closest_long <- makeGRangesFromDataFrame(closest_long,  start.field = "start", end.field = "end", ig
names(gr_closest_long) <- c(1:length(gr_closest_long))
```

Annotate the TSS

```
listEnsemblArchives()
```

```
##              name     date                                  url version
## 1  Ensembl GRCh37 Feb 2014          https://grch37.ensembl.org  GRCh37
## 2     Ensembl 107 Jul 2022 https://jul2022.archive.ensembl.org     107
## 3     Ensembl 106 Apr 2022 https://apr2022.archive.ensembl.org     106
## 4     Ensembl 105 Dec 2021 https://dec2021.archive.ensembl.org     105
## 5     Ensembl 104 May 2021 https://may2021.archive.ensembl.org     104
## 6     Ensembl 103 Feb 2021 https://feb2021.archive.ensembl.org     103
## 7     Ensembl 102 Nov 2020 https://nov2020.archive.ensembl.org     102
## 8     Ensembl 101 Aug 2020 https://aug2020.archive.ensembl.org     101
## 9     Ensembl 100 Apr 2020 https://apr2020.archive.ensembl.org     100
## 10     Ensembl 99 Jan 2020 https://jan2020.archive.ensembl.org      99
## 11     Ensembl 98 Sep 2019 https://sep2019.archive.ensembl.org      98
## 12     Ensembl 97 Jul 2019 https://jul2019.archive.ensembl.org      97
## 13     Ensembl 96 Apr 2019 https://apr2019.archive.ensembl.org      96
## 14     Ensembl 95 Jan 2019 https://jan2019.archive.ensembl.org      95
## 15     Ensembl 94 Oct 2018 https://oct2018.archive.ensembl.org      94
## 16     Ensembl 93 Jul 2018 https://jul2018.archive.ensembl.org      93
## 17     Ensembl 92 Apr 2018 https://apr2018.archive.ensembl.org      92
## 18     Ensembl 91 Dec 2017 https://dec2017.archive.ensembl.org      91
## 19     Ensembl 90 Aug 2017 https://aug2017.archive.ensembl.org      90
## 20     Ensembl 80 May 2015 https://may2015.archive.ensembl.org      80
## 21     Ensembl 77 Oct 2014 https://oct2014.archive.ensembl.org      77
## 22     Ensembl 75 Feb 2014 https://feb2014.archive.ensembl.org      75
## 23     Ensembl 54 May 2009 https://may2009.archive.ensembl.org      54
##    current_release
## 1
## 2               *
## 3
## 4
## 5
```

```
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
```

```
mart <- useMart(biomart = "ensembl", dataset = "mmusculus_gene_ensembl", host = "https://sep2019.archive
annoDataMart <- getAnnotation(mart, featureType = "TSS")
```

Annotate the TSS

```
gr_closest_long_anno <- annotatePeakInBatch(gr_closest_long,
                                AnnotationData=annoDataMart,
                                featureType = "TSS",
                                output="nearestLocation",
                                PeakLocForDistance = "start")

gr_closest_long_anno <- as.data.frame(gr_closest_long_anno)
```

Import the gene expression data

```
getwd()
```

```
## [1] "/Users/christiansom/Documents/GitHub/MAFLD_ER_agonists"
```

```
source("code/00_helper_functions.R")
symbol_geneID <- read.delim("data/ensembl_mmus_sep2019_annotation.tsv")[,1:2]

raw_counts <- read.table(
  file = 'data/bulkRNAseq_mmus_rawcounts.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::select(-PPT_HFD_male_4) %>%
  tibble::column_to_rownames('geneID') %>%
```

```
  as.matrix()

gene_len <- read.table(
  file = 'data/bulkRNAseq_mmus_gene_lengths.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE)

TPM <- normalizeData(x=raw_counts, len = gene_len$length, method = "TPM") %>%
  tibble::rownames_to_column("ensembl_gene_id")

TPM <- TPM %>%
  dplyr::select(ensembl_gene_id, CD_male_1, CD_male_4, HFD_male_2,HFD_male_1,DPN_HFD_male_1, DPN_HFD_ma
TPM <- inner_join(symbol_geneID, TPM, by="ensembl_gene_id")

# We name the mice according to their original mouse number instead of replicate number.
# CD2 and CD9 correspond to CDm1 and CDm4, HFD3 and HFD4 correspond to HFDm2 and HFDm1, DPN2 and DPN3 c

colnames(TPM) <- c("ensembl_gene_id", "symbol", "CDm2", "CDm9", "HFDm3", "HFDm4","DPN2", "DPN3", "E2_8"

gr_closest_long_anno_closest_genes <- gr_closest_long_anno %>%
  filter(!query_ID=="closest_ori") %>%
  dplyr::rename("ensembl_gene_id"="feature")

gr_closest_long_anno_closest_genes <- as.data.frame(gr_closest_long_anno_closest_genes)

TPM_filt <- TPM %>%
  dplyr::filter(ensembl_gene_id%in%gr_closest_long_anno_closest_genes$ensembl_gene_id)

chrom_TPM <- inner_join(gr_closest_long_anno_closest_genes, TPM_filt, by= "ensembl_gene_id") # 2 entrie

chrom_TPM2 <- chrom_TPM %>%
  dplyr::select("loc_ID" ,"seqnames", "start", "end", "enha.ID", "query_ID", "symbol", "ensembl_gene_id
```

IMPORT ENHANCER COUNTS and normalize table

```
library(dplyr)
library(tidyr)

counts_enha <- read.delim("results/Epigenome_analysis/diffbind_enhancers_1816_H3K27ac.clean.readCount",
names(counts_enha) <- c("CDm2_K27ac","CDm9_K27ac","HFDm3_K27ac","HFDm4_K27ac", "DPN2_K27ac","DPN3_K27ac
colsums_enha <- colSums(counts_enha[,])

counts_enha_norm <- sweep(counts_enha, 2, colsums_enha, FUN = "/")
counts_enha_norm2 <- counts_enha_norm *10^6
colSums(counts_enha_norm2[,])
```

```
##  CDm2_K27ac  CDm9_K27ac HFDm3_K27ac HFDm4_K27ac  DPN2_K27ac  DPN3_K27ac
##       1e+06       1e+06       1e+06       1e+06       1e+06       1e+06
##  E2_8_K27ac  E2_9_K27ac
##       1e+06       1e+06
```

```
counts_enha_norm2.1 <- counts_enha_norm2 %>% rownames_to_column("loc_ID")
K27_GE_joined <- inner_join(chrom_TPM2, counts_enha_norm2.1, by="loc_ID")
View(K27_GE_joined)
```

Subset the enhancer table and put into long format

```
sub_GE.K27_GE_joined <- K27_GE_joined %>%
  dplyr::select("loc_ID", "query_ID","symbol", "CDm2","CDm9","HFDm3","HFDm4", "DPN2","DPN3","E2_8","E2_
sub_GE.K27_GE_long <- pivot_longer(sub_GE.K27_GE_joined, cols=4:11, values_to = "Gene_expression")

sub.K27_K27_GE_joined <- K27_GE_joined %>%
  dplyr::select("loc_ID", "query_ID", "ensembl_gene_id","CDm2_K27ac","CDm9_K27ac","HFDm3_K27ac","HFDm4_
sub.K27_K27_GE_long <- pivot_longer(sub.K27_K27_GE_joined, cols=4:11, values_to = "H3K27ac")

K27_GE_long <- cbind(sub_GE.K27_GE_long, sub.K27_K27_GE_long)

K27_GE_long_dd <- K27_GE_long[!duplicated(as.list(K27_GE_long))]

# Remove the zeros to not correlate zeros (gives error message - but these genes are removed later anyh
K27_GE_long_dd <- K27_GE_long_dd %>%
  group_by(loc_ID, symbol) %>%
  mutate(filter_zeros = mean(Gene_expression)) %>%
  filter(filter_zeros > 0) %>%
  dplyr::select(!filter_zeros)
```

Import the reverted gene sets and filter the tables

```
K27_GE_long_group <- K27_GE_long_dd %>% group_by(loc_ID, query_ID) %>%
  mutate(correlation_pearson = cor(Gene_expression, H3K27ac, method="pearson")) %>%
  mutate(correlation_spearman = cor(Gene_expression, H3K27ac, method="spearman"))

write.table(K27_GE_long_group, "results/Epigenome_analysis/K27_GE_corr_before_filtering.txt", quote=F, 

DEGsets <- readRDS("results/bulkRNAseq_mmus_DEG_sets.rds")
revALL <- DEGsets$gene_id$reverted
length(revALL)
```

```
## [1] 379
```

```
K27_GE_long_rev_insect <-  K27_GE_long_group %>%
  filter(ensembl_gene_id %in% revALL)
length(unique(K27_GE_long_rev_insect$ensembl_gene_id))
```

```
## [1] 107
```

```
K27_GE_long_group_plot_pearson <- K27_GE_long_rev_insect %>%
  filter(abs(correlation_pearson) > 0.75) %>%
  group_by(loc_ID, query_ID) %>%
  mutate(name.ident = paste0(symbol, "_", loc_ID))

K27_GE_long_group_plot_pearson_revCount <- K27_GE_long_group_plot_pearson %>%
  mutate(name.ident = paste0(loc_ID, ":", symbol))

K27_GE_long_group_plot_spearman <- K27_GE_long_rev_insect %>%
  filter(abs(correlation_spearman) > 0.75) %>% group_by(loc_ID, query_ID) %>%
  mutate(name.ident = paste0(symbol, "_", loc_ID))

K27_GE_long_group_plot_pearson_before_Rev <- K27_GE_long_group %>%
  filter(abs(correlation_pearson) > 0.75) %>% group_by(loc_ID, query_ID) %>%
  mutate(name.ident = paste0(symbol, "_", loc_ID))


K27_GE_long_group_plot_spearman_before_Rev <- K27_GE_long_group %>%
  filter(abs(correlation_spearman) > 0.75) %>% group_by(loc_ID, query_ID) %>%
  mutate(name.ident = paste0(symbol, "_", loc_ID))
```

#Add 50kb to intersect CTCF peaks with the H3K27ac peaks

```
K27_GE_long_group_coordinates <- inner_join(K27_GE_long_group_plot_pearson, location, by="loc_ID")
K27_GE_long_group_coordinates$end <- as.integer(K27_GE_long_group_coordinates$end)

K27_GE_long_group_coordinates_left <- K27_GE_long_group_coordinates %>%
  dplyr::filter(query_ID=="closest_left1") %>%
  mutate(new_end = end+50000) %>%
  mutate(new_start=start)

K27_GE_long_group_coordinates_right <- K27_GE_long_group_coordinates %>%
  dplyr::filter(query_ID=="closest_right1" | query_ID=="closest_right2") %>%
  mutate(new_start = start-50000) %>%
  mutate(new_end=end)

K27_GE_long_group_coordinates_left_export <- K27_GE_long_group_coordinates_left %>%
  dplyr::select("chrom", "new_start", "new_end", "loc_ID", "query_ID", "symbol") %>%
  unique()

K27_GE_long_group_coordinates_right_export <- K27_GE_long_group_coordinates_right %>%
  dplyr::select("chrom", "new_start", "new_end", "loc_ID", "query_ID", "symbol") %>%
  unique()

write.table(K27_GE_long_group_coordinates_left_export, "results/Epigenome_analysis/H3K27ac_left_non_int
write.table(K27_GE_long_group_coordinates_right_export, "results/Epigenome_analysis/H3K27ac_right_non_i

#From here, intersect the CTCF peaks with the exported H3K27ac regions using BEDtools (command line)
```

prepare the CTCF files - separate by motif-orientation.

```
#load the motif-discovery file of CTCF motifs in mm10 genome by FIMO
FIMO_CTCF <- read.delim("data/fimo_mm10_genome_CTCFscan.tsv", sep="\t")

FIMO_CTCF_plus_bed <- FIMO_CTCF %>%
  dplyr::filter(strand=="+") %>%
  dplyr::select("chrom"="sequence_name", "start", "end"="stop", "strand")

FIMO_CTCF_minus_bed <- FIMO_CTCF %>%
  dplyr::filter(strand=="-") %>%
  dplyr::select("chrom"="sequence_name", "start", "end"="stop", "strand")

write.table(FIMO_CTCF_plus_bed, "results/Epigenome_analysis/fimo_mm10_genome_CTCF_plus.bed", quote=F, r
write.table(FIMO_CTCF_minus_bed, "results/Epigenome_analysis/fimo_mm10_genome_CTCF_minus.bed", quote=F,
```

HERE, RUN THE SHELL SCRIPT "Epigenome_06.03_CTCF_script_bedtools_enh_intersect.sh"

#after BEDTools intersection of H3K27ac enhancers (that have a good correlation with nearby genes) with nearby CTCF peaks, re-import

```
library(tidyverse)

names <- c("chrom", "start", "end", "loc_ID", "query_ID", "symbol")
H3K27ac_left_CTCFx_outwards <- read.delim("results/Epigenome_analysis/H3K27ac_left_CTCF.intersect.noncan

H3K27ac_left_CTCFx <- read.delim("results/Epigenome_analysis/H3K27ac_left_CTCF.intersect.canon.uniq.bed

H3K27ac_right_CTCFx_outwards <- read.delim("results/Epigenome_analysis/H3K27ac_right_CTCF.intersect.non

H3K27ac_right_CTCFx <- read.delim("results/Epigenome_analysis/H3K27ac_right_CTCF.intersect.canon.uniq.b
```

#combine these data.frames, because they comprise the enhancer-gene pairs that we can report

```
H3K27ac_CTCF_intersect <- rbind(H3K27ac_left_CTCFx_outwards, H3K27ac_left_CTCFx, H3K27ac_right_CTCFx_ou
table(H3K27ac_CTCF_intersect$CTCF_pos)
```

```
##
## canonical  outwards
##        50        33
```

```
length(unique(H3K27ac_CTCF_intersect$loc_ID))
```

```
## [1] 67
```

```
unique_symbols <- unique(H3K27ac_CTCF_intersect$symbol)
length(unique_symbols)
```

```
## [1] 45
```

```
# 45 unique genes that underlie potential enhancer-mediated estrogen-dependent regulation
```

#Compare the fold-change values for these sites - in addition to the reads in peaks this gives information about how much these enhancers are changed

```
CDvsHFD_H3K27ac_Diffbind <- readRDS("results/Epigenome_analysis/Diffbind_results_FDR_fold.rds")$all_DB_
 mutate(loc_ID = paste0(seqnames, ".",start,".",end), .before = seqnames) %>%
  dplyr::select(loc_ID, Fold, FDR)

# These are the enhancers intersected with CTCF. But more informative with foldchanges from Diffbind. T
H3K27ac_CTCF_intersect_log2FC <- inner_join(H3K27ac_CTCF_intersect,CDvsHFD_H3K27ac_Diffbind, by="loc_ID

# Retrieves the unique person corr values from BEFORE the CTCF intersection
corr_values <- K27_GE_long_group_plot_pearson %>%
  ungroup() %>%
  dplyr::select("loc_ID", "correlation_pearson", "symbol") %>%
  unique() %>% group_by(symbol)

# Creates a dataframe between the corr values and log2FCs. This one is the COMPLETE dataframe
H3K27ac_CTCF_intersect_log2FC_corr <- inner_join(corr_values, H3K27ac_CTCF_intersect_log2FC, by=c("loc_
duplicated(H3K27ac_CTCF_intersect_log2FC_corr)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
length(unique(H3K27ac_CTCF_intersect_log2FC_corr$symbol))
```

```
## [1] 45
```

```
# This unique ID is necessary to join the data frames in the next section. Otherwise, enhancers that ar
H3K27ac_CTCF_intersect_log2FC_corr <- H3K27ac_CTCF_intersect_log2FC_corr %>%
  mutate(name.ident = paste0(loc_ID, ":", symbol))

# To plot, we need the single columns for gene expression and log2FC again.  Note: some enhancers have
K27_GE_long_group_plot_filt <- K27_GE_long_group_plot_pearson %>%
  group_by(symbol, loc_ID) %>%
  mutate(name.ident = paste0(loc_ID, ":", symbol)) %>%
  filter(name.ident%in%H3K27ac_CTCF_intersect_log2FC_corr$name.ident)

K27_GE_long_group_plot_filt <- K27_GE_long_group_plot_filt[!duplicated(K27_GE_long_group_plot_filt), ]
length(unique(K27_GE_long_group_plot_filt$symbol))
```

```
## [1] 45
```

```
# The following should yield "character(0)"
setdiff(K27_GE_long_group_plot_filt$symbol, H3K27ac_CTCF_intersect_log2FC_corr$symbol)
```

```
## character(0)
```

```
length(unique(K27_GE_long_group_plot_filt$symbol)) # 45 unique genes
```

```
## [1] 45
```

```
nrow(K27_GE_long_group_plot_filt)/8 # 68 enhancer - gene pairs
```

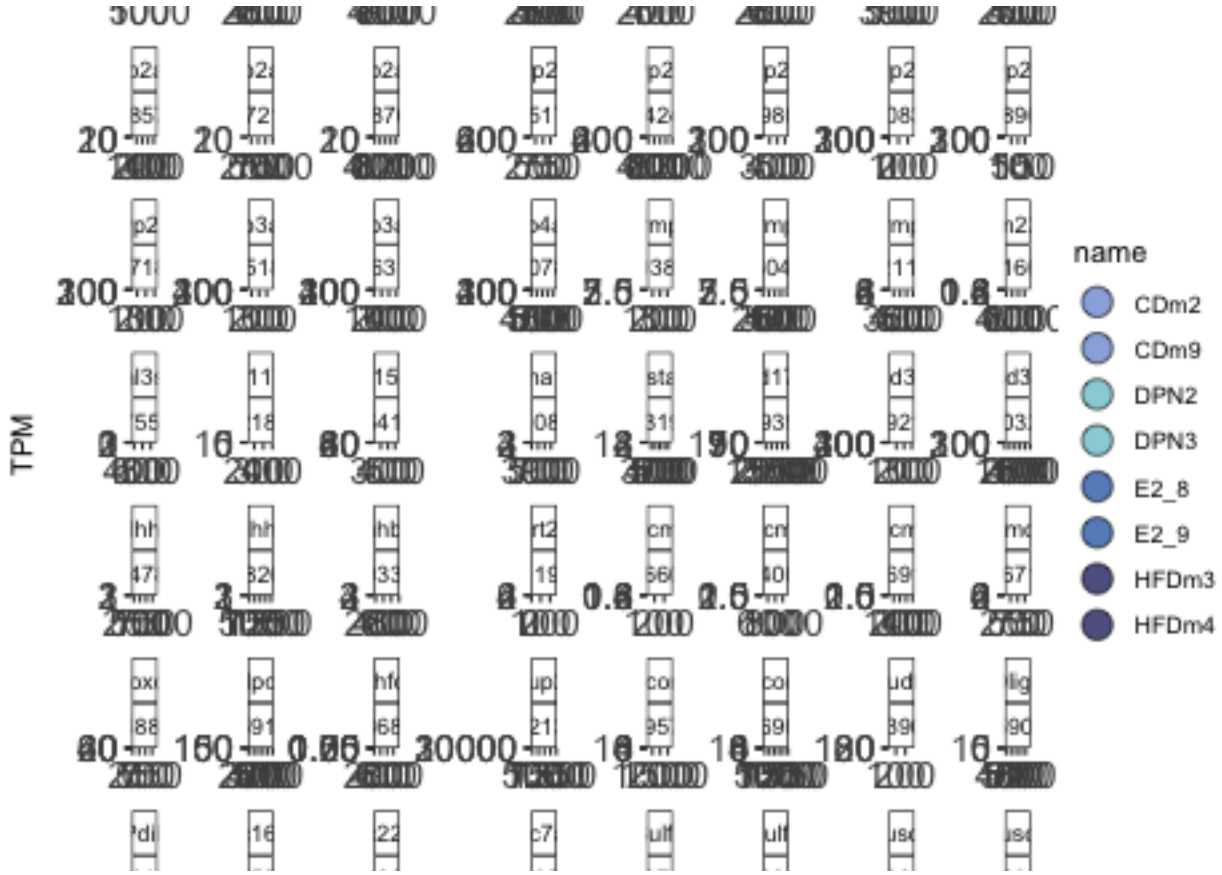
```
## [1] 68
```

```
length(unique(K27_GE_long_group_plot_filt$loc_ID)) # 67 unique enhancers
```

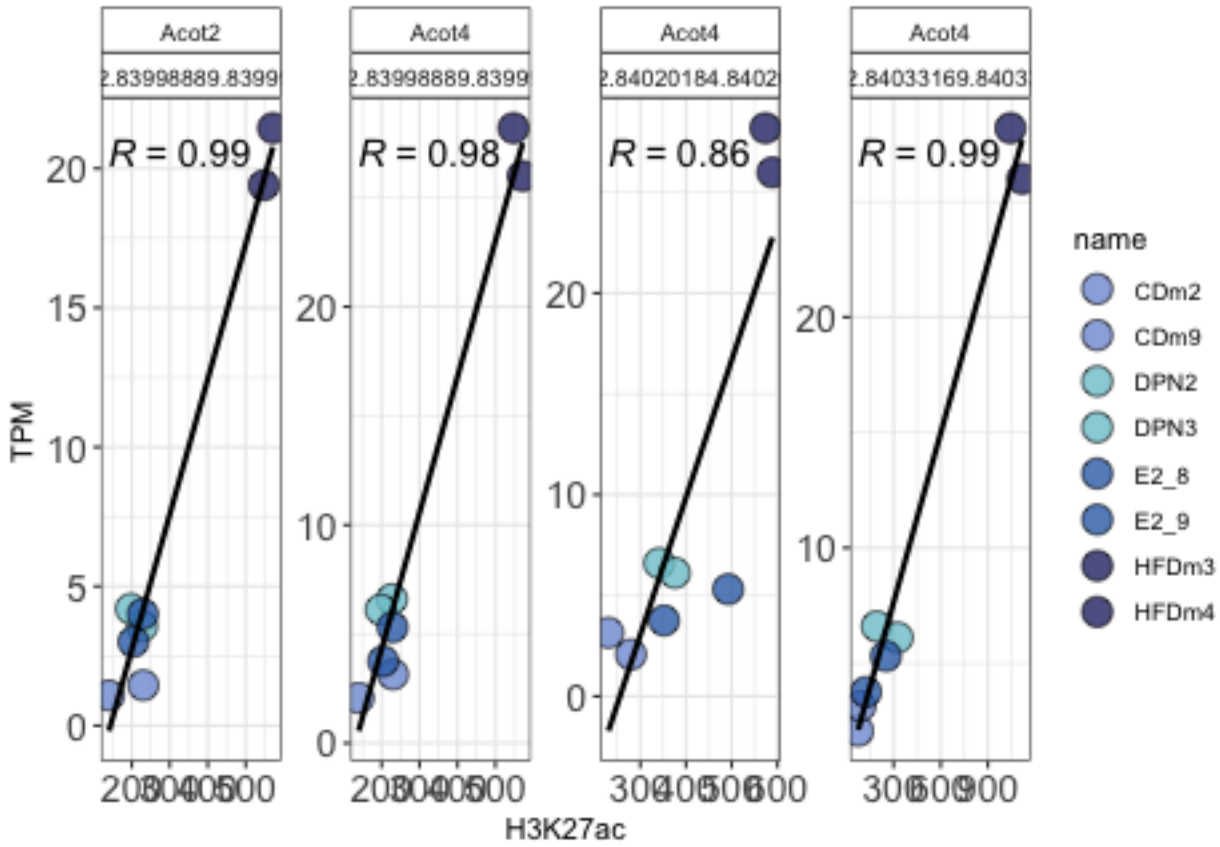
```
## [1] 67
```

```
# 68 total combinations of location IDs and genes, one enhancer goes for to genes.

write.table(K27_GE_long_group_plot_filt, "results/Epigenome_analysis/corr_45genes_67enh_toPlot.txt", ro
```

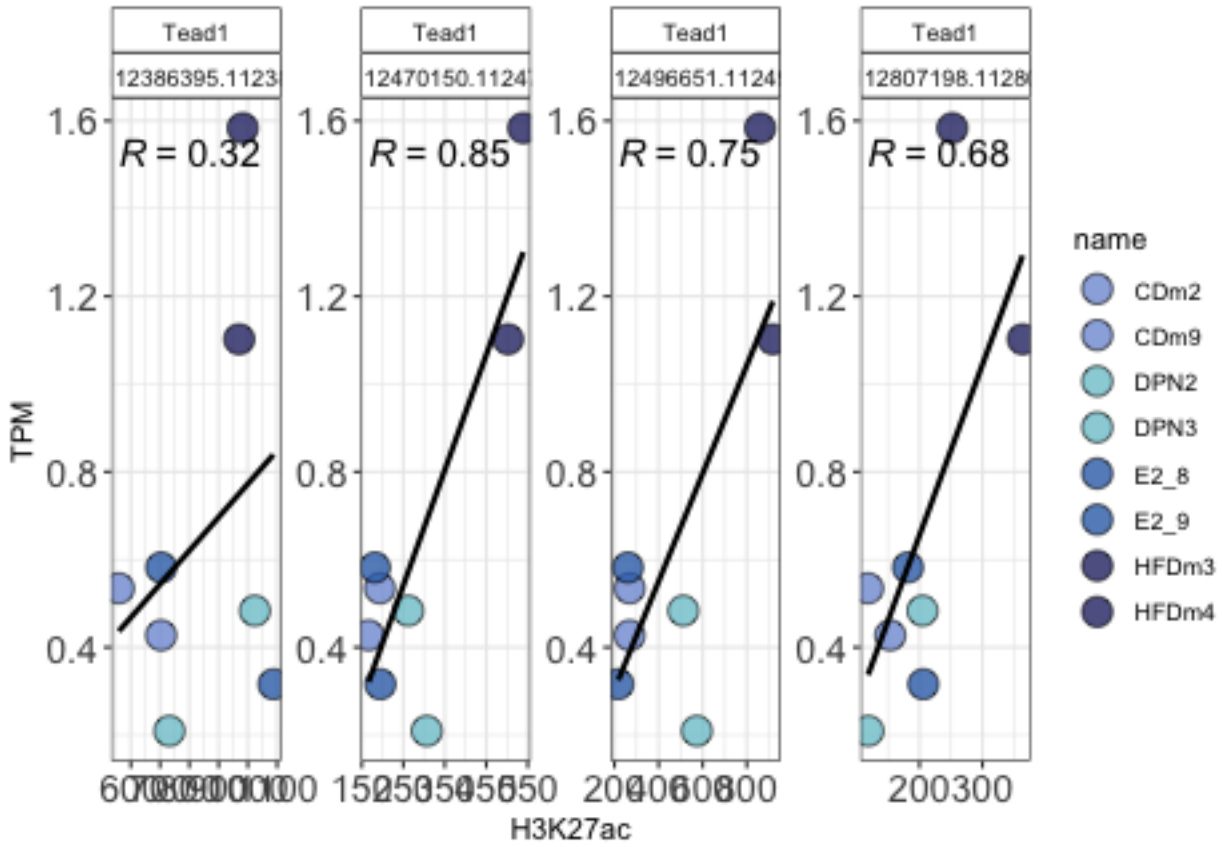
```
library(ggpubr)
  ggscatter(K27_GE_long_group_plot_filt, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            # yscale = "log2",
            # xscale = "log2",
            xlab="H3K27ac",
            ylab="TPM",
            palette=c("#7f9ad7","#7f9ad7", "#7dc7d1", "#7dc7d1", "#356fb5","#356fb5","#2c2f72","#2c2f72
    stat_cor(aes(label = ..r.label..),method="pearson", p.digits=0, size=5) +
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales = "free", ncol=8)
```



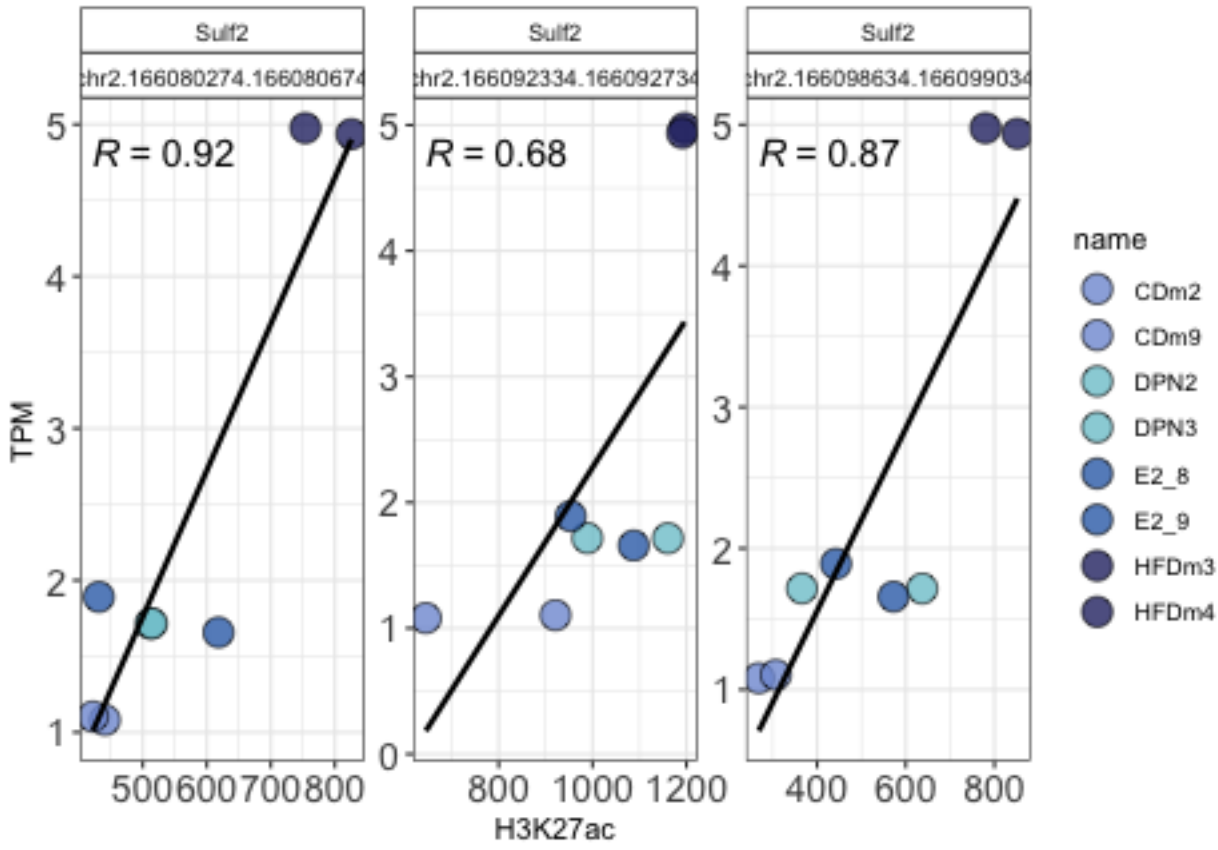

```
Acot_filt <- K27_GE_long_group_plot_filt %>% dplyr::filter(grepl("Acot", symbol))
ggscatter(Acot_filt, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            # yscale = "log2",
            # xscale = "log2",
            xlab="H3K27ac",
            ylab="TPM",
            palette=c("#7f9ad7","#7f9ad7", "#7dc7d1", "#7dc7d1", "#356fb5","#356fb5","#2c2f72","#2c2f72
    stat_cor(aes(label = ..r.label..),method="pearson", p.digits=0, size=5) +
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales="free", ncol=4)
```




```
# plot Tead1
tead1 <- K27_GE_long_group %>% dplyr::filter(symbol=="Tead1") %>% filter(!loc_ID=="chr7.112688427.11268
# note: this loc_ID is filtered out because it does not have a CTCF peak closeby (at least not a signif
#tead1_filt <- K27_GE_long_group_plot_filt %>% dplyr::filter(symbol=="Tead1")  # Plot this to only show
ggscatter(tead1, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            # yscale = "log2",
            # xscale = "log2",
            xlab="H3K27ac",
            ylab="TPM",
            palette=c("#7f9ad7","#7f9ad7", "#7dc7d1", "#7dc7d1", "#356fb5","#356fb5","#2c2f72","#2c2f72
    stat_cor(aes(label = ..r.label..),method="pearson", p.digits=0, size=5) +
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales="free", ncol=4)
```




```
# One enhancer has a correlation of 0.67 and hence does not pass the filter. I think this enhancer is i
Sulf2_also_non_filter_pass <- K27_GE_long_group %>% dplyr::filter(symbol=="Sulf2") %>% filter(!loc_ID==
Sulf2_all_enhancers <- K27_GE_long_group %>% dplyr::filter(symbol=="Sulf2")
#Sulf2 <- K27_GE_long_group_plot_filt %>% dplyr::filter(symbol=="Sulf2")  # Plot this to only show E-G-p
ggscatter(Sulf2_also_non_filter_pass, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            # yscale = "log2",
            # xscale = "log2",
            xlab="H3K27ac",
            ylab="TPM",
            palette=c("#7f9ad7","#7f9ad7", "#7dc7d1", "#7dc7d1", "#356fb5","#356fb5","#2c2f72","#2c2f72
    stat_cor(aes(label = ..r.label..),method="pearson", p.digits=0, size=5) +
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales="free", ncol=4)
```



Plot histogram to show in which processes (of the 24 GSEA) the 45 genes fall into.

```
library(clusterProfiler)
K27_GE_long_group_plot_filt <- read.delim("results/Epigenome_analysis/corr_45genes_67enh_toPlot.txt")
length(unique(K27_GE_long_group_plot_filt$symbol))
```

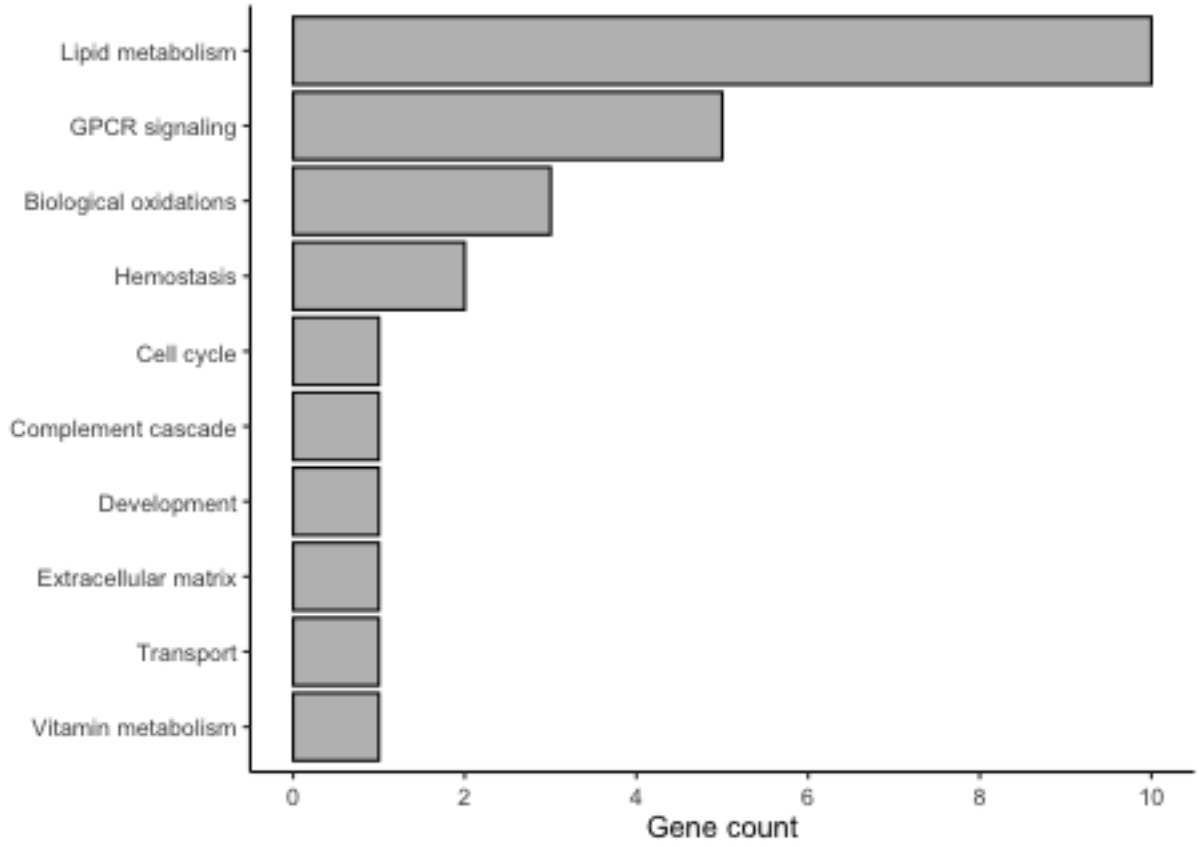
```
## [1] 45
```

```
symbols <- K27_GE_long_group_plot_filt$symbol %>% unique()

reactome_pathways <- readRDS("results/bulkRNAseq_mmus_GSEA_reactome_cluster_sets.rds")
reactome_pathways.2 <- as.data.frame(do.call(cbind, reactome_pathways))

reactome_pathways.long <- pivot_longer(reactome_pathways.2, cols=1:24)
intersect_pathways_symbols <- reactome_pathways.long %>% filter(value %in% symbols) %>% unique()
intersect_pathways_symbols_counts <- as.data.frame(table(intersect_pathways_symbols$name))

order.GSEA.pathways <- as.data.frame(table(intersect_pathways_symbols$name)) %>%
  arrange(-Freq) %>%
  dplyr::pull("Var1") %>%
  as.vector()
```

```
str(order.GSEA.pathways)
```

```
##  chr [1:10] "Lipid metabolism" "GPCR signaling" "Biological oxidations" ...
```

```
ggplot(intersect_pathways_symbols) +
  geom_histogram(aes(x=factor(name, levels=rev(order.GSEA.pathways))), stat="count", fill="grey", color=
  coord_flip() +
  theme_classic() +
  theme(axis.text.x = element_text(vjust = .5)) +
  xlab("") +
  ylab("Gene count") +
  scale_y_continuous(limits = c(), breaks=c(0,2,4,6,8,10))
```




```
library(clusterProfiler)
options(connectionObserver = NULL)
# Warning: call dbDisconnect() when finished working with a connection
library(org.Mm.eg.db)

unique_symbols <- K27_GE_long_group_plot_filt$symbol %>% unique()
length(unique_symbols)
```

```
## [1] 45
```

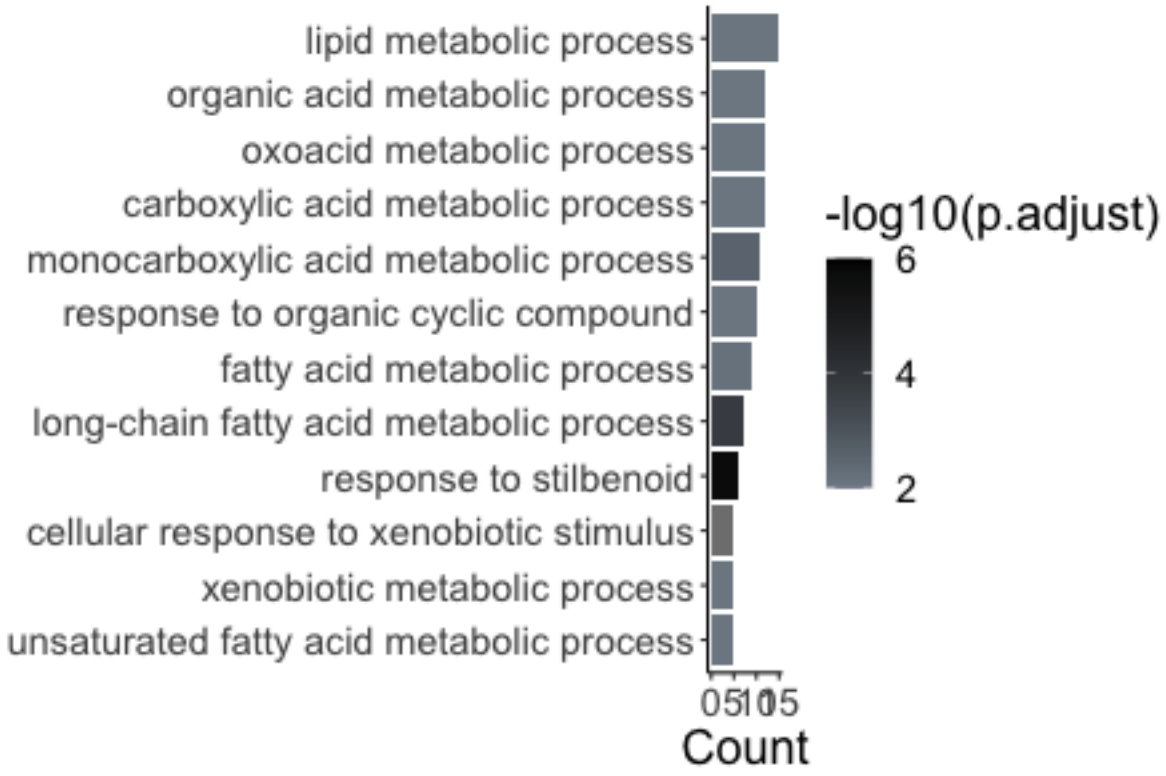
```
GO_BP <- enrichGO(gene = unique_symbols,
                  keyType       = 'SYMBOL',
                  OrgDb         = org.Mm.eg.db,
                  ont           = "BP",
                  pAdjustMethod = "BH",
                   pvalueCutoff  = 0.05,
                  qvalueCutoff  = 0.05,
                  minGSSize     = 1,
                  readable      = F,
                  universe = TPM_filt$symbol) # Can also use all expressed genes here instead.
head(as.data.frame(GO_BP))
```

```
##                    ID                                Description GeneRatio
## GO:0035634 GO:0035634                   response to stilbenoid      6/43
## GO:0001676 GO:0001676  long-chain fatty acid metabolic process      7/43
## GO:0032787 GO:0032787    monocarboxylic acid metabolic process     11/43
## GO:0006631 GO:0006631              fatty acid metabolic process      9/43
## GO:0033559 GO:0033559 unsaturated fatty acid metabolic process      5/43
## GO:0006805 GO:0006805              xenobiotic metabolic process      5/43
##             BgRatio       pvalue     p.adjust       qvalue
## GO:0035634   9/2674 9.785371e-10 1.691891e-06 1.465746e-06
## GO:0001676  29/2674 2.011636e-07 1.739060e-04 1.506610e-04
## GO:0032787 134/2674 4.827068e-06 2.782000e-03 2.410147e-03
## GO:0006631  97/2674 1.520198e-05 6.571056e-03 5.692742e-03
## GO:0033559  24/2674 2.873511e-05 8.166164e-03 7.074641e-03
## GO:0006805  25/2674 3.549299e-05 8.166164e-03 7.074641e-03
##                                                                        geneID
## GO:0035634                                 Slc22a7/Hsd3b5/Cyp2b9/Cyp2a5/Cd36/Gsta2
## GO:0001676                              Acot2/Cyp2b9/Cyp2a5/Cyp2a22/Acot4/Cyp4a10/Cd36
## GO:0032787 Mthfd1l/Acot2/Abcd2/Cyp2b9/Cyp2a5/Cyp2a22/Nudt7/Acot4/Mpc1/Cyp4a10/Cd36
## GO:0006631                 Acot2/Abcd2/Cyp2b9/Cyp2a5/Cyp2a22/Nudt7/Acot4/Cyp4a10/Cd36
## GO:0033559                                       Abcd2/Cyp2b9/Cyp2a5/Cyp2a22/Cyp4a10
## GO:0006805                                       Cyp2b9/Cyp2a5/Cyp2a22/Cyp3a11/Gsta2
##            Count
## GO:0035634     6
## GO:0001676     7
## GO:0032787    11
## GO:0006631     9
## GO:0033559     5
## GO:0006805     5
```

```
View(as.data.frame(GO_BP))
```

```
  library(dplyr)
  library(ggplot2)
  plot_me_ordered <- GO_BP[order(GO_BP$p.adjust), ]
  plot_me_ordered <- plot_me_ordered[1:12, ]
  plot_me_ordered <- plot_me_ordered[order(plot_me_ordered$Count), ]
  name_order <- plot_me_ordered %>%
    dplyr::pull("Description")

  ggplot(plot_me_ordered, aes(x=factor(Description, levels=name_order), fill=-log10(p.adjust), y=Count))
    geom_bar(stat="identity") +
```

```
    coord_flip() +
    scale_fill_gradient(low = "#808b96", high = "black",
    limits = c(2, 6), breaks = c(2, 4, 6))+
    theme_classic() +
    theme(text=element_text(size = 18)) +
    ggtitle("") +
    xlab("")
```




```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
```

```
## other attached packages:
##  [1] clusterProfiler_4.4.4 ggpubr_0.4.0          biomaRt_2.52.0
##  [4] org.Mm.eg.db_3.15.0   AnnotationDbi_1.58.0  Biobase_2.56.0
##  [7] ChIPpeakAnno_3.30.1   GenomicRanges_1.48.0  GenomeInfoDb_1.32.4
## [10] IRanges_2.30.1        S4Vectors_0.34.0      BiocGenerics_0.42.0
## [13] forcats_0.5.2         stringr_1.4.1         dplyr_1.0.9
## [16] purrr_0.3.4           readr_2.1.2           tidyr_1.2.0
## [19] tibble_3.1.8          ggplot2_3.3.6         tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.2                  tidyselect_1.1.2
##   [3] RSQLite_2.2.16              grid_4.2.1
##   [5] BiocParallel_1.30.3         scatterpie_0.1.7
##   [7] munsell_0.5.0               codetools_0.2-18
##   [9] withr_2.5.0                 colorspace_2.0-3
##  [11] GOSemSim_2.22.0             filelock_1.0.2
##  [13] highr_0.9                   knitr_1.40
##  [15] rstudioapi_0.14             ggsignif_0.6.3
##  [17] DOSE_3.22.1                 MatrixGenerics_1.8.1
##  [19] labeling_0.4.2              GenomeInfoDbData_1.2.8
##  [21] polyclip_1.10-0             bit64_4.0.5
##  [23] farver_2.1.1                downloader_0.4
##  [25] treeio_1.20.2               vctrs_0.4.1
##  [27] generics_0.1.3              lambda.r_1.2.4
##  [29] xfun_0.32                   BiocFileCache_2.4.0
##  [31] regioneR_1.28.0             R6_2.5.1
##  [33] graphlayouts_0.8.1          AnnotationFilter_1.20.0
##  [35] gridGraphics_0.5-1          bitops_1.0-7
##  [37] cachem_1.0.6                fgsea_1.22.0
##  [39] DelayedArray_0.22.0         assertthat_0.2.1
##  [41] BiocIO_1.6.0                scales_1.2.1
##  [43] ggraph_2.0.6                enrichplot_1.16.2
##  [45] googlesheets4_1.0.1         gtable_0.3.0
##  [47] tidygraph_1.2.2             ensembldb_2.20.2
##  [49] rlang_1.0.4                 splines_4.2.1
##  [51] rtracklayer_1.56.1          rstatix_0.7.0
##  [53] lazyeval_0.2.2              gargle_1.2.0
##  [55] broom_1.0.0                 yaml_2.3.5
##  [57] reshape2_1.4.4              abind_1.4-5
##  [59] modelr_0.1.9                GenomicFeatures_1.48.3
##  [61] backports_1.4.1             qvalue_2.28.0
##  [63] RBGL_1.72.0                 tools_4.2.1
##  [65] ggplotify_0.1.0             ellipsis_0.3.2
##  [67] RColorBrewer_1.1-3          Rcpp_1.0.9
##  [69] plyr_1.8.7                  progress_1.2.2
##  [71] zlibbioc_1.42.0             RCurl_1.98-1.8
##  [73] prettyunits_1.1.1           viridis_0.6.2
##  [75] SummarizedExperiment_1.26.1 haven_2.5.1
##  [77] ggrepel_0.9.1               fs_1.5.2
##  [79] magrittr_2.0.3              data.table_1.14.2
##  [81] futile.options_1.0.1        DO.db_2.9
##  [83] reprex_2.0.2                googledrive_2.0.0
##  [85] ProtGenerics_1.28.0         matrixStats_0.62.0
##  [87] patchwork_1.1.2             hms_1.1.2
```

```
##  [89] evaluate_0.16               XML_3.99-0.10
##  [91] VennDiagram_1.7.3           readxl_1.4.1
##  [93] gridExtra_2.3               compiler_4.2.1
##  [95] shadowtext_0.1.2            crayon_1.5.1
##  [97] htmltools_0.5.3             ggfun_0.0.6
##  [99] mgcv_1.8-40                 tzdb_0.3.0
## [101] aplot_0.1.6                 lubridate_1.8.0
## [103] DBI_1.1.3                   tweenr_2.0.1
## [105] formatR_1.12                dbplyr_2.2.1
## [107] MASS_7.3-57                 rappdirs_0.3.3
## [109] Matrix_1.4-1                car_3.1-0
## [111] cli_3.3.0                   parallel_4.2.1
## [113] igraph_1.3.4                pkgconfig_2.0.3
## [115] GenomicAlignments_1.32.1    xml2_1.3.3
## [117] InteractionSet_1.24.0       ggtree_3.4.2
## [119] multtest_2.52.0             XVector_0.36.0
## [121] rvest_1.0.3                 yulab.utils_0.0.5
## [123] digest_0.6.29               graph_1.74.0
## [125] Biostrings_2.64.1           rmarkdown_2.16
## [127] cellranger_1.1.0            fastmatch_1.1-3
## [129] tidytree_0.4.0              restfulr_0.0.15
## [131] curl_4.3.2                  Rsamtools_2.12.0
## [133] rjson_0.2.21                lifecycle_1.0.1
## [135] nlme_3.1-157                jsonlite_1.8.0
## [137] carData_3.0-5               futile.logger_1.4.3
## [139] viridisLite_0.4.1           BSgenome_1.64.0
## [141] fansi_1.0.3                 pillar_1.8.1
## [143] lattice_0.20-45             KEGGREST_1.36.3
## [145] fastmap_1.1.0               httr_1.4.4
## [147] survival_3.3-1              GO.db_3.15.0
## [149] glue_1.6.2                  png_0.1-7
## [151] bit_4.0.4                   ggforce_0.3.4
## [153] stringi_1.7.8               blob_1.2.3
## [155] memoise_2.0.1               ape_5.6-2
```